Building a Responsibility Model using Modal Logic - Towards Accountability, Capability and Commitment Concepts

Feltus, Christophe; Petit, Michaël

Published in:
Procedings of 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-2009), Rabat, Morocco

DOI:
10.1109/AICCSA.2009.5069353

Publication date:
2009

Document Version
Early version, also known as pre-print

Building a Responsibility Model using Modal Logic - Towards Accountability, Capability and Commitment Concepts

Christophe Feltus
PReCISE Researcher Centre
University of Namur, Belgium, and
Public Research Centre Henri Tudor
Luxembourg-Kirchberg, Luxembourg
christophe.feltus@tudor.lu

Michaël Petit
PReCISE Researcher Centre
Faculty of Computer Science
University of Namur, Belgium
mpe@info.fundp.ac.be

***Abstract*— This paper aims at building a responsibility model based on the concepts of Accountability, Capability and Commitment. This model's objective is, firstly, to help organizations verify the organization structure and detect policy problems and inconsistencies, and secondly, to provide a conceptual framework to support them in defining their corporate, security and access control policies. Our work provides a preliminary review of the research performed in that field and proposes, based on the observations, an UML responsibility model and a definition for all its components. Thereafter, we suggest and explain a deontological logic formalization of the most significant concepts. To achieve that, our innovation stands in the adaptation of the Traditional Threefold Classification from the alethic logic to an adapted threefold classification that targets "Responsibility" and based upon which commitment is somewhat refined.**

I. INTRODUCTION

It is remarkable that nowadays, the responsibility committed from a person to perform a task, an aspect that for a long time remained overshadowed, appears to be of major interest. This responsibility [18] and [19] is often perceived as a combination of rights and obligations. However, current business (for instance in the financial sector) demonstrates that the moral aspect is improvable, and that taking care of that matter would avoid, in some cases, malfunctions of the system. In practice, responsibility is most often translated through policies. Many definitions of policy exist. For our work, we prefer the definition of policies from [1] that Policies are rules governing the choice in behaviour of a system. This definition is interesting in that, even if it stems from a low level context, it sounds applicable to a high level one such as management.

Based upon the above observations, the objective of the paper is to propose a literature review of policy models and engineering methods to identify the main policy's concepts. From that literature review, a model of responsibility is elaborated and incorporates main responsibility concepts and the major relationships between these. This model aims to be generic enough to permit the declination of policies to all abstract layers of the company as well as policies compatible to all domains of application, e.g. a high level policy for the management up to a low level policy of access rights. Finally, we propose a formalization of the concepts using logic system. The main formalization objectives are, first, to propose a basic logic framework for defining all concepts and second, by using that framework, to verify the organisational structure and detecting policy problems and inconsistency.

Identify applicable sponsor/s here. *(sponsors)*

II. RESPONSIBILITY LITERATURE REVIEW

It is rapidly observable, when analyzing policy literature, that a very large number of authors show interest in that concern. Consequently, a number of surveys have already been produced in that domain [2][3][4] and [5] but none has targeted the responsibility through the triplet (Capability, Accountability, Commitment).

Despite that proliferation of works, it is noteworthy that up to now there a distinction between works addressing access control model, policy model, role engineering and permission/policy engineering does not really exist. Based on that assumption, it appears meaningful for apprehending that topic to clarify this point and to highlight the existing dichotomy between model and method. To perform our review, we will base our analysis on a commonly accepted idea that a model or conceptual model is a representation designed to show the structure of a system or concept and that (at least in our case), a method is a body of techniques for collecting data necessary to instantiate the conceptual model. Consequently and as illustration, the Role-Based Access Control (RBAC) model [6] proposes a structure for providing access based on role, whereas role engineering [7] and [8] is a method aiming to define roles to instantiate the conceptual model. Identically, policy may also be modelled, and there exists a proliferation of methods to instantiate it. These

methods may be classified according to the technique they use. We propose to start with methods based on Requirements Engineering (RE) and to continue with a list of others. Moreover, it is more frequent to read papers targeting policy language than policy model. Those policy languages are innumerable and spread over the entire organizational model layers. The most famous of them are Ponder [5], Policy Description Language [6], Security Policy Language [7], and Rei [8]. Amazingly, the policy model used to support the policy expression through the policy language remains rarely specified. This review successively presents the responsibility through access control models and engineering methods. The components of the responsibility's triplet are:

- Capability: which describes the quality of having the prerequisite qualities or accesses to resources to achieve a task;
- Accountability: which describes the state of being accountable on the achievement of a task;
- Commitment: which concerns the engagement of a stakeholder to fulfil a task, and the assurance he will do it.

These definitions are refined through the description of these concepts in section 4.

Responsibility in the field of IT has already been investigated because of IT security constraints and requirements firstly, and of software requirement engineering secondly. IT security depicts responsibility mainly when it addresses access control. Indeed, to provision employees with rights and obligations to operate an application or a component, main access control model use the concept of role for group employees based on their responsibility, function, geographic location, domain of work, etc. Some examples of those models are the Mandatory Access Control, RBAC [10], UCON [11], OrBAC [12], etc. However, the inconvenient already observed in large company is that the engineering of these roles sometimes leads to situations where the amount of roles is bigger than the amount of employees. This is summarized in Table I.

Responsibility has also been subject of research in the field of software requirement engineering. Indeed, this concept is concentric for a large amount of methods like I*[13]. I* makes goal-oriented strategic modelling and analysis of requirements by using three mains concepts that are: actors, intentional elements, and links. Actors are described in their organizational settings and have attributes such as goals, abilities, beliefs, and commitments. Actors can be agents, roles, and positions. Agents are concrete actors, systems or humans, with specific capabilities. The inconvenient of those methods is that they are limited to concepts directly linked to the software requirement like the right or the obligation without offering the possibility to be extended to wider concepts like the commitment.

The state of the art of policy concepts introduces a review of four main recognized access control models: Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role-based Access Control (RBAC) and Usage Control Model (UCON).

Our survey has also covered others approaches that, due to the size of the paper, are not presented here. In summary we may observe that firstly, some concepts are commonly accepted, such as right, role and obligation. Definitions of the two firsts concepts are scarce. Only one definition has been found for the concept of "right": the right (or permission) is explicitly granted to a subject to access an object in a specific mode, such as read or write [1]. For the concept of "role", only one definition has been found in [13]. The concept of obligation is subject to more debate. For Bettini et al. [14], obligations are conditions or actions that must be fulfilled either by the users or the system after a decision. In [1], Sandhu et al. define obligations as requirements that have to be fulfilled by the subject for allowing access. Crook et al. [15] extend the notion of obligation to obligation policies relating to actions that must be carried out on targets by subjects when a predefined event occurs, and Haley et al. in [16] define it as which actions must be taken before access can be granted.

TABLE I. AC MODEL AND RESPONSIBILITY'S CONCEPTS

	MAC	**DAC**	**RBAC**	**UCON**
Subject	Yes	Yes	Yes	Yes
Object	Yes	Yes	Yes	Yes
Group	No	User Group	Role	Defined by objects and subject's attributes
Capability	Access Right	Access Right	Access Right	Access Right
Accountability (Obligation, Constraint)	No	No	Yes, static and dynamic separation of duty	Defined by objects and subject's attributes
Commitment	No	No	No	No

Table II is a summary and a comparison of the reviewed engineering methods. We may observe that, because the most frequently addressed concern of capability is the access right, existing models and methods most of the time remain targeting low-level layers of abstraction of the organization. Moreover, if we consider responsibility as a tuple (Capability, Accountability, Commitment), we observe that nowadays no model and method exist that entirely take into account all these responsibility components.

TABLE II. ENGINEERING METHODS AND RESPONSIBILITY'S CONCEPTS.

	KAOS	I*	GBRAM	ARMF	RACAF	Scenario Driven	Uses Cases
Subject	Agent	Actors	Agent	Users	Actors	Subject	Actors
Object	Yes	Yes	-	Asset	Data	-	Object
Group	-	Yes	-	Yes	Yes	Yes	Yes
Capability (Right, Authorzation)	Authorization rules	Abilities and beliefs	-	Permission	Permission	Permission	Access right
Accountability (Obligation, Constraint)	Achieve requirements and expectations	Goal	Achieve a goal	Perform a task	Perform a task	Perform a scenario	Pre-conditions, post-conditions
Commitment	No	Yes	No	No	No	No	No

III. MODELING RESPONSIBILITY CONCEPT

At the top of the UML model (see Figure 1) is the organization. Organizations encompass employees (users) whose objectives are to perform tasks (or processes) by using resources. To facilitate administration, those users are often grouped together based on their profile. As previously explained in the literature review, the most famous type of classification is the role, but variations exist such as for example the team, the hierarchy, or some geographical constraints. Existing solutions most often limit the resources accessibility to an access right conferred to a role. Our model covers that possibility, but extends it to the notion of responsibility that also encompasses the accountability, the commitment, and the capability. In our model, capability is a broader concept than the mere one of access right. Our model encompasses the following concept:

User: a person, external or internal to an organization, that has to achieve a task s/he is responsible for;

Role: describes the position of a person in the organisation. This position may be related to a hierarchical status, a geographic position, the membership in an organisation unit or department, or whatever;

Resource: is something needed for or produced by performing a task. Resource can take different forms such like information, manpower;

Task: is the operation performed by the users;

Capability: describes the quality of having the required qualities, skills or resources to perform a task;

Accountability: describes the state of being accountable (responsible) on the achievement of a task;

Commitment: is the engagement of a stakeholder to fulfil a task and the assurance that s/he will do it;

Access right: is a statement on the type of action that could be performed by a user through a resource;

Organization: is an entity that encompasses users, resources and processes, that aims at pursues collective goals.

Our model reuses some commonly accepted components presented in the literature survey in sections 3 and 4, whereas others are new. User is the basic component and appears as a person, a system or a software component. Resource could take a large scale of representation. Capability is a component that is part of all models and methods. Capability is a component that is part of all models and methods. Capability is most frequently declined through definition of access rights, authorizations or permissions. Accountability is a component that exists mainly in engineering methods and that is the obligation to achieve a task or to perform an action. Commitment is the most infrequent concept. Traditional policy model such as RBAC do not address it, however i* partly introduces it (e.g. when defining dependency as an "agreement" between two actors), but knowing whether it is a moral concept or an obligation remains subject to interpretation.

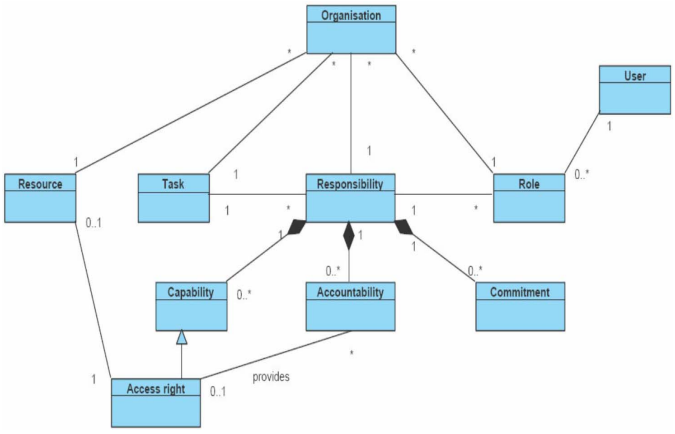

Figure 1. UML Model of Responsibility

In the first part of this section, we tackle at explaining main responsibility concepts. The second part of this section will put forward some significant relationships between them. As basic relation, the link between Role, Responsibility and Task is to be underlined. Indeed, it is no longer justified that the main construct of an organization is the performance of a task by an employee implicitly generating profit. Responsibility is the midpoint concept that lies down between Task and Employee (declined in the model under the concept of Role), and that adds essence to this relation. On this relation is to be read: "*there is one and only one role responsible for one task, and one role may have many responsibilities and one responsible may perform many tasks*". The second significant relation to be discussed in the paper is, subsequently, the relationship between Responsibility and Capacity, Accountability and Commitment. This relation is of the form

0..* to 1. That means that being responsible involves the possibility to dispose of many Capacities, Accountabilities and Commitment. But on the opposite, Commitment is only bound to one responsibility, and adequately for Accountability and Capability. The last quite interesting relation is the one that concerns the access to a Resource, and more precisely, the Access right to a Resource. This Access Right is a Capability for a person responsible, while being at the same time an Accountability for another.

IV. FORMALIZING CONCEPTS USING STANDARD AND DEONTIC LOGIC

Responsibility may be formalized using standard logic and more precisely SDL (for Standard Deontic Logic) theories. Standard Logic is the logic of necessary truth and related relations. This chapter attempts to define the responsibility (R) assigned to a user (u) to perform a task (t) and is written R([t]u). The responsibility is defined as according to the capability (CA), the accountability (AC) and the commitment (CO) as explained on Figure 1.

In [17], Cholvy et al. propose a formalization of the responsibility concept . To achieve this, they begin their work with a definition of the various meanings of responsibility, and then model several aspects of responsibility using SDL and logic of actions. The three concepts {capability, accountability and commitment} implicitly exist through the responsibility definitions, but are not duly modelled. For example, definition 2 issued from Cholvy's paper claims that responsibility is an obligation or a moral duty to report or explain the action, or someone else's action to a given authority (answerability). This definition helps at defining the commitment as a moral duty in parallel with an obligation that is a legal duty. Definition 3, which defines the responsibility according to a position in an organization, explains that someone responsible for something should be prepared to justify his action. This justification brings the content of the concept of accountability and consequently nuances accountability versus answerability. This definition also argues that it is possible to analyze the "consistency" of an organization by identifying users overloaded with responsibility. It brings up the notion of user capability in the sense of having enough resources to assume a number of responsibilities.

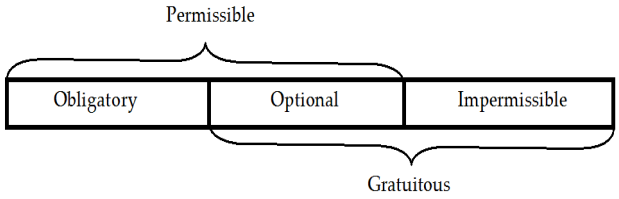

Figure 2. Traditional Threefold Classification

$$IMp \leftrightarrow OB\neg p \quad (1)$$

$$PEp \leftrightarrow \neg OB\neg p \quad (2)$$

$$GRp \leftrightarrow \neg OBp \quad (3)$$

$$OPp \leftrightarrow (\neg OBp \,\&\, \neg OB\neg p) \quad (4)$$

In addition to Cholvy's proposition to formalize responsibility with deontic logic, our work provided an adaptation of the traditional threefold classification (TTC) (Figure 2) on firstly transposing Obligatory by Accountable in that both bring up the notion of anything indispensable and makes obligatory through a legal issue (like a policy). Secondly, we transpose Impermissible by Incapable in that both defend the idea that it is not permitted or not allowable. Thirdly, an optional proposition of the deontic standard logic is analogue to an optional (OP) proposition in a responsibility based threefold classification. To achieve that transposition, we defined the incapacity (IN) and the unaccountability (UN) such as :

$$IN[t]u \leftrightarrow AC\neg[t]u \quad (5)$$

$$UN[t]u \leftrightarrow \neg AC[t]u \quad (6)$$

Equally to the deontic standard schema, the Figure 3 highlights that the three rectangular cells are jointly exhaustive and mutually exclusive.

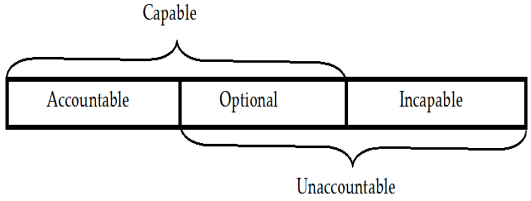

Figure 3. Responsibility based Threefold Classification

Indeed, each proposition is accountable, optional or incapable. Moreover, Capable propositions are those that are either accountable or optional and unaccountable propositions are those that are either optional or incapable. Moreover, we may define the capability and optional concept based on the accountability one such as:

$$CA[t]u \leftrightarrow \neg AC\neg[t]u \quad (7)$$

$$OP[t]u \leftrightarrow (\neg AC[t]u \,\&\, \neg AC\neg[t]u) \quad (8)$$

The equation (8) asserts that something is an optional proposition if and only if neither the performance of the task t by the user u nor its negation is accountable. This first assertion of the optional proposition issued from the adaptation of the deontological logic TTC is very interesting in that it expresses a first statement of being through the proposition of accountability. However, accountability is a proposition that aims to link two stakeholders: the accountable (or the responsible) and its manager (or the organization). It appears that this option proposition, even if optional to an organizational accountability, could remain engaged toward a moral obligation that we call Commitment. This commitment could be defined as the act of binding itself (intellectually or emotionally) to a course of actions. The set of commitment

possibilities in (9) aims at defining the optional proposition according to the commitment.

$$CO\ Type = \{CO[t]u \vee CO\neg[t]u \vee CO[\neg t]u \vee CO\neg[\neg t]u\} \quad (9)$$

This proposition that based on the user's (u) commitment for achieving the task t, 4 possibilities exist:

1. u is committed to achieve t;
2. u is committed not to achieve t;
3. u is committed to achieve not t;
4. u is committed not to achieve not t.

For achieving a task, u must have the necessary capabilities and be committed to perform it. Whether or not he is accountable do not presents any impact on the realization. Whatever, not achieving a task for which the user is accountable may lead to some kind of blame. This aspect is not discussed in that paper. Commitment is possible to be represented on Figure. 2 as shown on Figure 4.

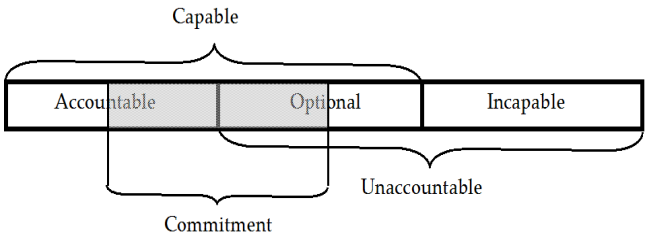

Figure 4. Commitment on Responsibility Threefold Classification

To accept a responsibility, a user must consequently have received necessary capability, he must also provide evidences to obligations he is accountable for and finally, he must be committed to perform obligations. Responsibility R is now definable using alethic and deontic logic on considering that:

1. The responsibility is for a user the obligation (noted □ with Classical ML) to perform a task and is equivalent to the accountability for that task (10)

$$R[t]u \rightarrow \Box[t]u \approx R[t]u \rightarrow \Box AC[t]u \quad (10)$$

2. To be responsible of performing a task, it is obliged that the user receives necessary capability. Providing these capability is another user's (u') responsibility (11 et 11').

$$R[t]u \rightarrow \Box CA[t]u \quad (11)$$

$$\approx$$

$$R[t]u \rightarrow R[\text{« provide [t]u capability »}]u' \quad (11')$$

Moreover, because user (u) capability (c) is received when user (u') the task ([« provide [t]u capability »]u'), we can state in (12)

$$\Box CA[t]u \rightarrow \Box\text{« provide [t]u capability »}]u \quad (12)$$

3. It is necessary that he provides commitment to perform that task (13):

$$R[t]u \rightarrow CO[t]u \quad (13)$$

Consequently, the responsibility may be formalized be the addition of (10), (11) and (13).

$$R[t]u \leftrightarrow \Box AC[t]u \wedge \Box CA[t]u \wedge CO[t]u \quad (14)$$

V. CASE STUDY

To illustrate the formal definition of the responsibility developed in the previous section, we propose the following little case study: "*Mister Boss is the manager of the marketing company named "SelltheWorld". Each year, Mister Boss organizes during the Christmas period a large mailing of postcards to all customers. This year, Mr Boss has too much work for closing the annual report and consequently decides to delegate this task to one of the employees. Because the task is less business sensitive as some other production task, Mr Boss decides to delegate it to a part-time secretary named Sophie. Sophie has just got married and consequently, she accepts this additional work without commitment. Mr Boss asks the IT service to give Sophie the necessary access right to the customers address list. John from this service realizes the necessary operation for providing this right as soon as he gets the request. On 30th January, Mister Boss receives over 100 complains from customers who didn't receive Christmas cards.*"

This case study permits to highlight responsibility toward the task of sending postcards:

Mr Boss has duly formalized Sophie's Accountability by asking her to process the sending activity. It was consequently clear what she was accountable to do. To achieve the mailing, she got the necessary capability that was the access to the customers file. However, due to the fact that her thought went to her new husband rather than to the work to accomplish, she didn't really want to achieve the work and failed to assure her responsibility due to a lack of commitment. Sophie's responsibility of sending postcard $\rightarrow 1 \wedge 2 \wedge 3$

1) Sophie's obligation of having the capacity to access customer file.

2) Sophie's obligation to get the accountability to achieve that task from Mr Boss

3) Sophie is committed to achieve it.

Responsibility could also not be assured in the case where she didn't get the necessary capabilities, i.e. if Mr Boss forgets to ask the IT service to provide the necessary access right or if the IT service didn't do as requested. No guarantee of having the job performed would also have been assured if Mr Boss had not clearly asked Sophie to send postcards. In that case, she would not have received the due Accountability.

John's responsibility can also be analyzed with that case study. John is a well paid IT staff who is very happy with his

function. He has received clear accountability to give access right to Sophie and he has the needed capabilities due to his position as network administrator. He has consequently been responsible to fulfil Mr Boss' request.

Sophie's obligation of having the capacity to access customer file $\rightarrow 4 \wedge 5 \wedge 6$

4. John's obligation of having the capacity to give access right to the customer file.
5. John's obligation to get the accountability to provide that access from Mr Boss.
6. John is committed to achieve it.

We may consequently observe that the responsibility of John to provide access to the customer file precedes the capability of Sophie to possess those rights.

VI. CONCLUSION

We have analyzed the literature to understand the semantics of AC policy conceptual models and engineering methods. We have observed that some elements are commonly accepted components whereas others remain debated or not addressed. Accepted concepts include the one of user (and related ones such as group or role), the one of resource and the one of Capability. Capability is most frequently delineated under access right, authorizations or permissions. Accountability is a concept that exists mainly in engineering methods and that is delineated as the obligation to achieve a task or to perform an action. Commitment is the most infrequent concept.

Based upon that observation, we have developed a conceptual model of responsibility as a UML class diagram. We have provided a definition for all the conceptual components and clarified some important relationships between those (relation task-responsibility-user, responsibility-capability-accountability-commitment and resource-capability-accountability). Based upon that model, we have proposed a formal description of the responsibility using modal deontological logic theory. Finally, a case study has been drawn to illustrate the whole idea.

In this paper, the responsibility concept has mainly by addressed based on an IT approach. The "organizational and management" field is also rich of responsibility's theory [34] and [35]. This area will be the focus of our future researches and will permit to refine our first findings. Consequently, our future works will focus on continuing the development of the model of responsibility, and most specifically the concept of commitment that is important to consider in high-level layers of the organizational model. Moreover, defining a policy that allows taking into account the commitment opens doors to new approaches that have right to be taken into account in traditional and renowned risk management solutions. Future investigations will e.g. deal with the case where the stakeholder commits to unmoral actions or actions that are different to the one requested (CO[¬t]u and CO¬[¬t]u).

Another part of our work aims at defining a new approach to propagate the responsibility from the high level down to the lower one. Our first researches demonstrate that potential solutions are to link responsibility concepts with an organization's processes. To support the progress of that approach, a software prototype has been developed based on "egroupware open framework". Those researches and the prototype have been presented in [19].

REFERENCES

[1] N. Dulay, E. Lupu, M. Solman, N. Damianou, A Policy Deployment Model for the Ponder Language ,International Symposium on Integrated Network Management, Seattle, May 2001, IEEE Press.

[2] Robert Crook, Darrel Ince, Bashar Nuseibeh, Modelling access policies using roles in requirements engineering, Information and Software Technology 45 (2003) 979-991.

[3] Robert Crook, Darrel Ince, Bashar Nuseibeh, On Modelling access policies: Relating Roles to their Organisational Context, RE 2005, Paris.

[4] P. A. Epstein, Engineering of Role/Permission Assignement, PhD thesis.

[5] R. Crook, D. Ince, B. Nuseibeh, "Using i* to Model Access Policies: Relating Roles to their Organisational Context", Social Modelling for Requirements Engineering,, MIT Press, 2006

[6] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn and R. Chandramouli, Proposed NIST Standard for Role-Based Access Control, ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, Pages 224-274

[7] G. Neumann, M. Strembeck, A Scenario-driven Role Engineering Process for Functional RBAC Roles, SACMAT'02, June 34, 2002, Monterey, California, USA.

[8] E. J. Coyne,. 1996. Role engineering. First ACM Workshop on Role-Based Access Control, Gaithersburg, Maryland, United States.

[9] N. Damianou, N. Dulay, E. Lupu, M. Sloman , The Ponder Policy Specification Language Workshop on Policies for Distributed Systems and Networks (Policy2001), HP Labs Bristol, 29-31. Springer-Verlag.

[10] E. Bertino, A. Mileo, A. Provetti, PDL with Preferences. IEEE international Workshop on Policies For Distributed Systems and Networks, Policy 2005 – Vol. 00, IEEE Computer Society, Washington, DC, 213-222.

[11] C. Basile, A. Lioy, G. Perez, C. Martinez; F. J. Garcia; Skarmeta, A. F. Gomez, POSITIF: A Policy-Based Security Management SystemPolicies for Distributed Systems and Networks, 2007. POLICY'07, pp. 280 – 280.

[12] K. Lalana, Rei : A Policy Language for the Me-Centric Project, TechReport, HP Labs, September 2002.

[13] R. Sandhu, J. Park, Usage Control: A Vision for Next Generation Access Control, The Second International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, 2003

[14] A. Miège, Définition d'un environnement formel d'expression de politiques de sécurité. Modèle Or-BAC et extensions. Thèse Sécurité informatique, INFRES, Télécom Paris [ENST] (2005)

[15] E. S. Yu, L. Liu, 2001. Modelling Trust for System Design Using the i* Strategic Actors Framework. Workshop on Deception, Fraud, and Trust in Agent Societies Held During the Autonomous, Eds. Lecture 35 194

[16] C. Bettini, S. Jajodia, X. S. Wang, D. Wijesekera, Provisions and Obligations in Policy Management and Security Applications, 28th VLDB conference, China, 2002

[17] L. Cholvy, F. Cuppens, and C. Saurel. Towards a logical formalization of responsibility. In Proc. of the Sixth International Conference on Artificial Intelligence and Law, pages 233--242, 1997

[18] C. Feltus, A. Rifaut, An Ontology for Requirements Analysis of Managers' Policies in Financial Institutions, I-ESA2007, Portugal

[19] B. Gâteau, C. Feltus, J. Aubert, C. Incoul, An Agent-based Framework for Identity Management: The Unsuspected Relation with ISO/IEC 15504, IEEE RCIS 2008, 3-6/6/2008, Marrakech, Morocco.